Synopsis on

"Some Aspects of Machine Learning applications in Power Systems"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF
DOCTOR OF PHILOSOPHY
IN
ELECTRICAL ENGINEERING

Submitted By
Puneet Aggarwal
Roll No. 11139526
Registration No. 08-ECM-3496

UNDER THE GUIDANCE OF
Dr. S. K. Aggarwal
(Professor& HEAD)

ELECTRICAL ENGINEERING DEPARTMENT
MAHARISHI MARKANDESHWAR ENGINEERING COLLEGE
MAHARISHI MARKANDESHWAR UNIVERSITY
MULLANA, AMBALA

Electrical Engineering Department
M. M. Engineering College
M. M. University, Mullana

Dated: 07-08-14

DRC meeting was conducted to consider and approve the thesis synopsis of Mr. Puneet Aggarwal (Research Scholar, EED)

Name of Scholar	:	Puneet Aggarwal
Registration No.	:	08-ECM-3496
Title of Thesis	:	Some Aspects of Machine Learning applications in Power Systems
Name of Supervisor	:	Dr. S. K. Aggarwal

Thesis title of Mr. Puneet Aggarwal is approved and recommended to put up before Board of Studies (EE) and then faculty of Engineering (MMEC) for further consideration and approval.

Members:

Dr. S. K. Aggarwal

Mrs. Ashu Ahuja

Mr. Aman Ganesh

Received

INDEX

Sr. No.	Contents	Page No.
1.	Abstract	I
2.	Introduction	1
3.	Literature Survey	6
4.	Motivation Of work	11
5.	Problem Formulation and the Proposed work	12
6.	Methodology	13
7.	Time Line	24
8.	References	25

ABSTRACT

Machine learning (ML) is a part of Artificial Intelligence that deals with the study of algorithms and systems that can learn from the data they process& analyze. ML has found its use in many applications ranging from self-driving cars, facial recognition and speech recognition etc. ML also have many applications in power system as well, which are energy monitoring, energy disaggregation problem, renewable power predictions, cost optimization, price and load forecasting, time series predictions& power system security and maintenance. The objective of current research is to apply ML tools such as ANN, SVM etc. on power system (or energy related) problems as: energy disaggregation problem, load forecasting problem, price forecasting problem, wind & solar power prediction problems and to measure the accuracy of the various tools applied.

INTRODUCTION

Machine learning is a part of artificial intelligence (AI) that deals with the study of systems that can learn from data. For example, a machine learning system could be trained on email messages to learn to distinguish between Primary, Social and Promotion messages. After learning, it can then be used to classify new email messages into Primary, Social and Promotion folders. [1]

The main body of machine learning deals with Representation, Evaluation and Optimization. Representation of data means that the data can be easily understandable and can be easily trained by all machine learning systems. An Evaluation means that the objective function should be such that it can distinguished between the good classifiers from the bad ones. Finally Optimization means the method we use for searching among the various classifiers that gives the best result or the highest scoring one. The choice of optimization technique is key to the efficiency of the learner, and also helps determine the classifier produced if the evaluation function has more than one optimum. There are a wide variety of machine learning tasks and successful applications. Optical character recognition, in which printed characters are recognized automatically based on previous examples, is a classic example of machine learning. Machine learning focuses on prediction, based on *known* properties learned from the training data or Test Data.

Machine learning algorithms can be organized into a taxonomy based on the desired outcome of the algorithm or the type of input available during training the machine.

- *Supervised learning* algorithms are trained on *labelled* examples, i.e., input where the desired output is known. The supervised learning algorithm attempts to generalize a function or mapping from inputs to outputs which can then be used speculatively to generate an output for previously unseen inputs.
- *Unsupervised learning* algorithms operate on *unlabeled* examples, i.e., input where the desired output is unknown. Here the objective is to discover structure in the data (e.g. through a cluster analysis), not to generalize a mapping from inputs to outputs.
- *Semi-supervised learning* combines both labeled and unlabeled examples to generate an appropriate function or classifier.
- *Reinforcement learning* is concerned with how intelligent agents ought to act in an environment to maximize some notion of reward. The agent executes actions which cause the observable state of the environment to change. Through a sequence of actions, the agent attempts to gather knowledge about how the environment responds to its actions, and attempts to synthesize a sequence of actions that maximizes a cumulative reward.
- Developmental learning, elaborated for robot learning, generates its own sequences (also called curriculum) of learning situations to cumulatively acquire repertoires of novel skills through autonomous self-exploration and social interaction with human teachers, and using guidance mechanisms such as active learning, maturation, motor synergies, and imitation. [1]

Various machine learning tools are:

Decision Tree Learning

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. More descriptive names for such tree models are classification trees or regression trees. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making.

Artificial Neural Networks

An artificial neural network (ANN) learning algorithm, usually called "neural network" (NN), is a learning algorithm that is inspired by the structure and functional aspects of biological neural networks. Computations are structured in terms of an interconnected group of artificial neurons, processing information using a connectionist approach to computation. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs, to find patterns in data, or to capture the statistical structure in an unknown joint probability distribution between observed variables.

Support Vector Machines

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.

Bayesian Networks

A Bayesian network, belief network or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independencies via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Efficient algorithms exist that perform inference and learning.

Applications of machine learning in Power System Analysis:

1. Energy Disaggregation Problem

Imagine an energy feedback system that displays not only your total power consumption, but also continuously shows real-time usage, broken down by electrical appliance. Such a system could provide personalized and cost-effective energy saving recommendations. For example, it could report, "Based on your usage patterns, you could save $215 per year by switching to a more efficient heating unit, which will pay for itself in 27 months." The challenge in this scenario is to sense end-uses of energy to provide feedback at the fine-grained, appliance level.

There has been substantial prior research in this area, however most of this work has concentrated on the use of power consumption patterns and using changes in power draw as features to identify what appliance is being used and how much energy it is consuming.

A more recent approach to estimate appliance usage is to examine the Electromagnetic Interference (EMI) that most consumer electronic appliances produce as identifying signatures. This EMI is measured using a special sensor built at the Ubicomp Lab at the University of Washington. The plot is in frequency domain and shows the signatures of various appliances.

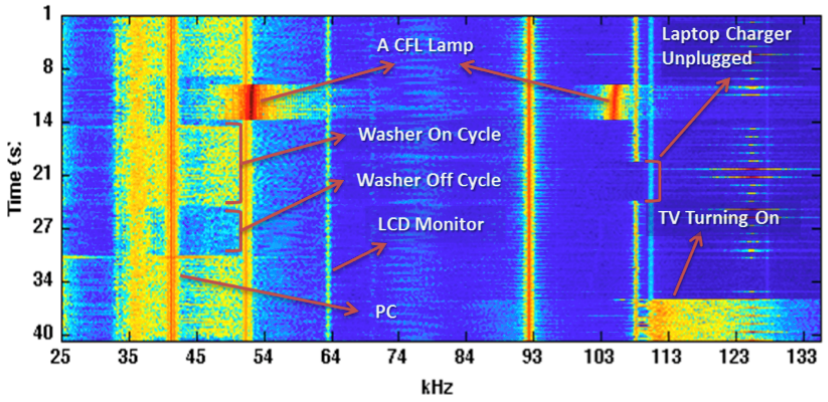

Fig. 1 Signatures of various appliances

The presence or absence of such EMI signatures would ideally tell us when a particular appliance is in use. However, due to the large numbers of appliances in a home, the solution is not straightforward. Machine learning is required not only to make an inference about the appliance class given a particular signature, but probabilistic models are needed that take into account, for example, human appliance usage patterns (think using coffee machine and toaster in morning vs. lights in evening), weather patterns (very unlikely that AC came on during winters), and appliance electrical model. The signature of an appliance can also drift or vary over time due to operating conditions and the mode in which they are used (for instance, a washing machine has many modes). [2]

2. Load Forecasting Problem

Load Forecasting (LF) provides a structured interface for creating, managing and analyzing load forecasts. Accurate models for electric power load forecasting are essential to the operation and planning of a utility company. DLF helps an electric utility to make important decisions including decisions on purchasing electric power, load switching, as well as infrastructure development.

Load forecasting is classified in terms of different planning durations: short-term load forecasting or STLF (up to 1 day, medium-term load forecasting or MTLF (1 day to 1 year), and long-term load forecasting or LTLF (1-10years). To forecast load precisely throughout a year, various external factors including weathers, solar radiation, population, per capita gross domestic product seasons and holidays need to be considered. For example, in the winter season, average wind chill factor could be added as an explanatory variable in addition to those used in the summer model. In transitional seasons such as spring and fall, the transformation technique can be used. For holidays, a holiday effect load can be deducted from the normal load to estimate the actual holiday load better.

Various predictive models have been developed for load forecasting based on various techniques like multiple regression, exponential smoothing, iterative reweighted least-squares, adaptive load forecasting, stochastic time series, fuzzy logic, neural networks and knowledge based expert systems.

3. Price Forecasting

Electricity pricing (sometimes referred to as electricity tariff or the price of electricity) varies widely from country to country, and may vary significantly from locality to locality within a particular country. There are many reasons that account for these differences in price. The price of power generation depends largely on the type and market price of the fuel used, government subsidies, government and industry regulation, and even local weather patterns. Electricity price forecasting is simply the process of using mathematical models to predict what electricity prices will be in the future. The simplest model for day ahead forecasting is to ask each generation source to bid on blocks of generation and choose the cheapest bids. If not enough bids are submitted, the price is increased. If too many bids are submitted the price can reach zero or become negative. The offer price includes the generation cost as well as the transmission cost along with any profit. Power can also be sold or purchased from adjoining power pools.

The fuel used to generate electricity at a power plant is the primary cost incurred by electrical generation companies. Particularly, coal, as a fuel for base load plants and more important, to a degree, natural gas for peaking plants affect power prices. This will change as more renewable energy is used, when the capital cost will be the primary cost, as renewable energy (other than biomass and biofuel) has no fuel cost.

4. Wind and Solar power forecasting.

A wind power forecast corresponds to an estimate of the expected production of one or more wind turbines (referred to as a wind farm) in the near future. By production is often meant available power for wind farm considered (with units kW or MW depending on the wind farm nominal capacity). Forecasts can also be expressed in terms of energy, by integrating power production over each time interval. In the electricity grid at any moment balance must be maintained between electricity consumption and generation - otherwise disturbances in power quality or supply may occur. Wind generation is a direct function of wind speed and, in contrast to conventional generation systems, is not easily dispatch able. Fluctuations of wind generation thus receive a great amount of attention. Variability of wind generation can be regarded at various time scales. First, wind power production is subject to seasonal variations, i.e. it may be higher in winter in Northern Europe due to low-pressure meteorological systems or it may be higher in summer in the Mediterranean regions owing to strong summer breezes. There are also daily cycles which may be substantial, mainly due to daily temperature changes. Finally, fluctuations are observed at the very short-term scale (at the minute or intra-minute scale). The variations are not of the same order for these three different timescales. Managing the variability of wind generation is the key aspect associated to the optimal integration of that renewable energy into electricity grids.

Solar power forecasting involves knowledge of the Sun´s path, the atmosphere's condition, the scattering processes and the characteristics of a solar energy plant which utilizes the Sun's energy to create solar power. Solar photovoltaic systems transform solar energy into electric power. The power output depends on the incoming radiation and on the solar panel characteristics. Photovoltaic power production is increasing nowadays. Forecast information is essential for an efficient use, the management of the electricity grid and for solar energy trading. The energy generation forecasting problem is closely linked to the problem of *weather variables forecasting*. Indeed, this problem is usually split into two parts, on one hand focusing on the forecasting of solar PV or any other meteorological variable and on the other hand estimating the amount of energy that a concrete power plant will produce with the estimated meteorological resource. In general, the way to deal with this difficult problem is usually related to the spatial and temporal scales we are interested in, which yields to different approaches that can be found in the literature. In this sense, it is useful to classify these techniques depending on the forecasting horizon, so it is possible to distinguish between *now-casting* (forecasting 3–4 hours ahead), *short-term forecasting* (up to 7 days ahead) and *long-term forecasting* (months, years…)

LITERATURE SURVEY

In literature review various machine learning tools are used for the solution of power systems applications described as under

- Energy Disaggregation Problem.
- Power System Security Problem.
- Load Forecasting Problem.
- Electricity Price Forecasting Problem.
- Smart Home Energy Management.
- Forecasting of Wind Power Generation.
- Solar Power Predictions.

Various machine learning tools applied to power system application in literature are as under:

- Particle Swarm Optimization.
- SVM, Improved SVM and Multiclass SVM.
- Extreme Learning Machine. (ELM)
- Artificial Neural Network.
- Fuzzy Logic.
- Bayesian Network.

The detailed literature is given as under:

Sr. No.	Ref. No.	Author	Year	Problem & objective		Approach
1	[3].	Berges, M., Goldman, E., Matthews, H.S., and Soibelman, L.	2008	electricity metering	collection of data	Hardware based sub metering Whole house continuous monitoring Non-intrusive load monitoring
2	[4].	Carrie Armel, K., Gupta, A., Shrimali, G., and Albert, A.	2012	Save energy by using smart meters	collect Appliance Specifies energy data	Smart meters, Sensing Technology, Smart Appliances etc.
3	[5].	Froehlich, J., Larson, E., Gupta, S., Cohn, G., Reynolds, M., and Patel, S.	2011	Disaggregated Energy Sensing	Collecting Disaggregated Data from 14 houses	Distributed Direct Sensing, Single Point Sensing, **Electrical Energy Disaggregation using Voltage noise**
4	[6].	Gupta, S., Reynolds, M., and Patel, S.	2010	Energy Monitoring	sensing Disaggregated electricity usage in the home	Electric sense senses the EMI), a solution for automatically detecting and classifying the use of electronic devices in home (Data collected from 7 homes)
5	[7].	Laughman, C., Lee, K., and Cox, R.	2003	Monitoring electrical appliances, detection of transients and harmonics.	NILM is used for extracting usefull at information about systems that uses electromechanical devices.	Non-intrusive load monitoring system, steady state monitoring techniques.
6	[8].	Zeifman, M., Ph, D., and Roth, K.	2011	Monitoring electrical appliances, detection of transients, harmonics etc.	collection of data or features of appliances using NIALM	non-intrusive appliance load monitoring

7	[9].	Rudin, C.; Waltz, D.; Anderson, R.N.; Boulanger, A; Salleb-Aouissi, A; Chow, M.; Dutta, H.; Gross, P.N.; Huang, B.; Ierome, S.; Isaac, D.F.; Kressner, A; Passonneau, R.J.; Radeva, A; Wu, L.	2012	Machine learning for preventive maintenance	To predict the risk of failures for components and systems.	Mean Time Between Failure, cross validation and blind test, state of art of machine learning algorithms
8	[10].	Hassan, T.; Javed, F.; Arshad, N.,	2014	energy disaggregation and monitoring	predictions of appliance labels	feed forward neural network, supervised learning, support vector machine
9	[11].	Figueiredo, M.; Ribeiro, B.; de Almeida, A,	2014	measuring the electrical consumption of individual appliances in a house hold	Energy disaggregation or load monitoring	non negative tensor factorization, Sparse coding
10	[12].	Qinran Hu; Fangxing Li,	2013	to reduce total electricity payment	Control the price for energy, sensors to detect human activities.	Smart home energy management system, machine learning to reduce price for energy
11	[13].	Srinivasarengan, K.; Goutam, Y.G.; Chandra, M.G.; Kadhe, S.,	2013	electricity metering only active powers	collection of data	single point measurement, unsupervised learning, Bayesian Network
12	[14].	Chaouachi, A; Kamel, R.M.; Andoulsi, R.; Nagasaka, K.,	2013	predict 24 h ahead photovoltaic generation and 1 h ahead wind power generation and LD	Multiobjective optimization, operating cost low,	Artificial Intelligence, Fuzzy logic, Neural network
13	[15].	Hongming Yang; Pandharipande, A,	2012	estimation of day light level	to sense Light output from light source and artificial light	estimation algorithms are performed, coherent and incoherent algorithms are developed
14	[16].	Mishra, A; Irwin, D.; Shenoy, P.; Kurose, J.; Ting Zhu,	2011	lower the electric bills without requiring consumers	To predict a model which forecasts future demand.	Smart charge, machine learning algorithms etc.

				involment		
15	[17].	Niyato, D.; Lu Xiao; Ping Wang,	2010	Home energy Management system	Minimize the cost of HEMS, collect status and power consumptions demand from home appliances.	Machine to machine communications, dynamic programing algorithm developed.
16	[18].	Filippi, A; Pandharipande, A; Lelkens, A; Rietman, R.; Schenk, T.; Ying Wang; Shrubsole, P.,	2010	Energy Monitoring	Activeness of the appliances, Power consumptions	Multi Appliance Power Disaggregation, Linear detection algorithm, NILM
17	[19].	Dhiman, G.; Rosing, T.S.,	2006	power consumptions	large power benefits	Dynamic power Management
18	[20].	Xia Chen; Zhao Yang Dong; Ke Meng; Yan Xu; Kit Po Wong; Ngan, H. W.,	2012	Electricity Price forecasting	nonlinear Modeling,	Extreme Learning Machine, Bootstrapping
19	[21].	Bo-Juen Chen; Ming-Wei Chang; Chih-Jen Lin,	2004	Load Forecasting	Predict load for the next 31 days.	Support Vector Machine, Regression, Time Series.
20	[22].	Can Wan; Zhao Xu; Pinson, P.; Zhao Yang Dong; Kit Po Wong,	2014	Wind Power Forecast	Predictions intervals are included.	Extreme Learning Machine based probalistic forecasting method is used.
21	[23].	Sharma, N.; Sharma, P.; Irwin, D.; Shenoy, P.,	2011	Predicting Renewable power generations	Solar power generation from weather predictions	Machine Learning, Support Vector Machine
22	[24].	Jinyu Tian; Yan Lin,	2009	Price Forecasting	used PSO for optimization	Rough Sets and Improved SVM
23	[25].	Fan, S.; Mao, C.; Chen, L.,	2007	Electricity Price forecasting	day ahead prediction	Self-organized map Hybrid Network, SVM
24	[26].	Yaochu Jin; Sendhoff, B.,	2008	cost optimization		EMO, ML, Multiobjective learning, Neural Network
25	[27].	Sapankevych, N.I; Sankar, Ravi,	2009	Time series predictions	Financial Market, Load forecasting, Price Forecasting , weather forecasting	Support vector machine, Time Series.

26	[28].	Sarpeshkar, R.; Wattanapanitch, W.; Rapoport, B.I; Arfin, S.K.; Baker, M.W.; Mandal, S.; Fee, M.S.; Musallam, S.; Andersen, R.A,	2007	Prosthetics, wireless neuroscience		neural simulation, analog and digital designs
27	[29].	Kalyani, S.; Swarup, K.S.,	2011	Security of electric power system	Static security, transient security	sequential forward selection method, multi class SVM
28	[30].	Haoyang Shen; Hino, H.; Murata, N.; Wakao, S.,	2011	Solar power prediction is reliable or not	Measure of Credibility	Shannon information content
29	[31].	Jiang Chang; Shu-Yun Jia,	2008	Wind solar power prediction		Multi agent system
30	[32].	Can Wan; Zhao Xu; Pinson, P.; Zhao Yang Dong; Kit Po Wong,	2014	Predictions intervals of wind power		Machine Learning, particle swarm optimization
31	[33].	Yan Xu; Zhao-Yang Dong; Zhao Xu; Ke Meng; Kit Po Wong,	2012	Security of electric power system	wind power	Extreme learning Machine, soft computing
32	[34].	Yan Xu; Zhao Yang Dong; Jun Hua Zhao; Pei Zhang; Kit Po Wong,	2012	Security of electric power system	Dynamic Security assessment.	Extreme learning Machine

Brief summary of Literature Survey

1. In references [3] – [8], [10] – [11], [13], [18], [19] the main focus of the authors is to collect data of appliances by detecting the Electromagnetic Inferences (EMI) of the appliances by using various hardware and software tools such as:

- Using Smart meters.
- Digital meters.
- Single point sensing device which sense EMI.
- Non-intrusive load monitoring system (NILMS) etc.

2. In references [14], [22], [23], [30] – [32] the authors predict the solar and wind power for short durations by using Machine Learning, PSO, ANN etc.
3. In references [15] – [17] the main focus of the authors is to reduce the consumption of the electricity in the houses by using different hardware and software tools.

4. In references [20], [21], [24], [25] price and load forecasting is done for short durations by using Extreme learning machine, Support Vector Machine etc.
5. In references [33] – [34] the authors applied Extreme learning Machine, soft computing for the protection and maintenance and security of Electric power system.

Motivation of the Work

From the literature review it is concluded that Energy Disaggregation and Electricity Price Forecasting are the challenging areas in the power system and there is a vast scope of research in these areas. Many machine learning tools have already been applied by various authors which give motivation for the application of ANN, Decision Tree and SVM and their comparative analysis on the above said problems.

Problem Formulation and the Proposed Work

The current research focus on power system problems as:

- **Energy disaggregation problem**

Energy Disaggregation problem is divided in two parts as

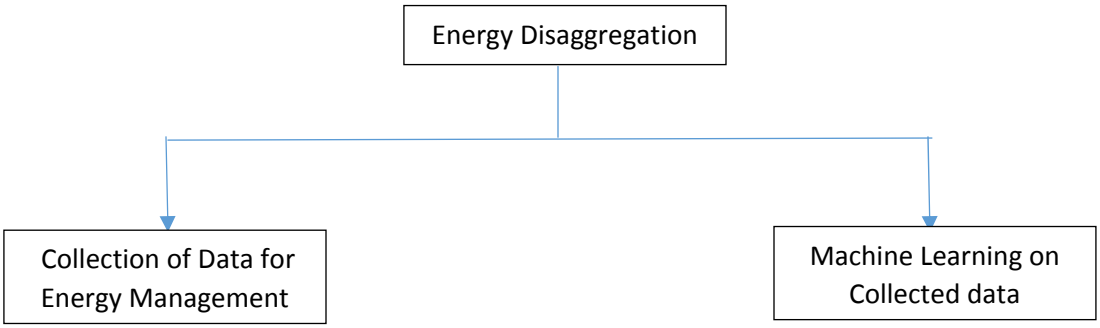

Fig. 2 Parts of energy Disaggregation

In the first part the data is collected by detecting the EMI of the Appliances at a particular place whether it is a house or industry or any other place where the energy is consumed. The data collected is in the form as shown below:

- Appliance ID (say 1, 3, 6, 30 etc. According to type)
- Appliance Name (Outside Over Garage Lights, Outside Front Door Lights, Downstairs Hallway Lights , Stairway Lights , Dining Room Lights etc.)
- Start Time. (Particular Appliance is ON)
- Stop Time. (Particular Appliance is OFF)

For every instance of time ranging from 12:00 AM to 11:59 PM. The data is collected for about six months to several months. As the time period is quite large a huge data is collected. For the collection of data machine learning is used for the collection of data.

The focus is on the publically or freely availability of data. Appliance ID, Appliance Name and a Particular time Stamp would be given. The aim of the work is to predict whether the appliance is ON or OFF at that particular instance of time. The problem is a multi-label classification problem because the appliances are large or different. For this multi label classification problem machine learning is required. Different machine learning tools would be referred for this multi label classification problem and compare their results for better accuracy.

- **Electricity price forecasting problem.**

  Electricity price forecasting for short duration would be done by various machine learning tools and the tools would be compared on the basis of minimum error in the predicted price.

METHODOLOGY USED

All the machine learning tools are implemented in Matlab Environment

I. **Artificial Neural Network :**

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

Neural networks, with the remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyses. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

1. **Adaptive learning:** Ability to learn how to do tasks based on the data given for training or initial experience.
2. **Self-Organisation:** An ANN can create its own organisation or representation of the information it receives during learning time.
3. **Real Time Operation:** ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. **Fault Tolerance via Redundant Information Coding:** Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Mathematical Model of a Neuron:

A neuron is an information processing unit that is fundamental to the operation of a neural network. The three basic elements of the neuron model are:

1. A set of weights, each of which is characterized by a strength of its own. A signal x_j connected to neuron k is multiplied by the weight wk_j. The weight of an artificial neuron may lie in a range that includes negative as well as positive values.
2. An adder for summing the input signals, weighted by the respective weights of the neuron.
3. An activation function for limiting the amplitude of the output of a neuron. It is also referred to as squashing function which squashes the amplitude range of the output signal to some finite value.

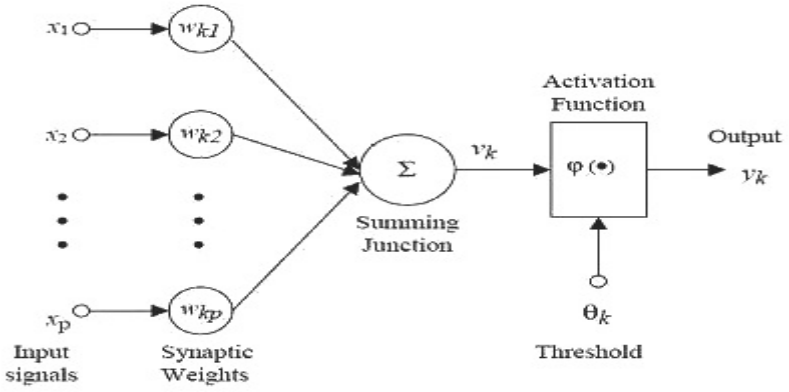

Fig. 3 Model of ANN

Model of ANN consists of:

1. A set of weights, each of which is characterized by a strength of its own.
2. An adder for summing the input signals, weighted by the respective weights of the neuron.
3. An activation function for limiting the amplitude of the output of a neuron.

$$Vk = \sum_{j=1}^{p} (Wkj\ Xj) \qquad \text{and} \quad Y_k = \varphi\ (V_k + \theta_k)$$

Network Architecture

There are three fundamental different classes of network architectures:

1) **Single-layer Feed Forward Networks**

In a layered neural network the neurons are organized in the form of layers. In the simplest form of a layered network, we have an input layer of source nodes that projects onto an output layer of neurons, but not vice versa. This network is strictly a Feed forward type. In single-layer network, there is only one input and one output layer. Input layer is not counted as a layer since no mathematical calculations take place at this layer.

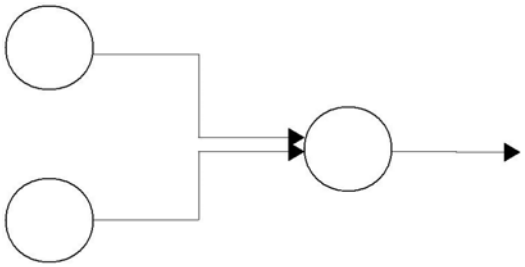

Fig. 4 Single-layer Feed Forward Network

2) **Multilayer Feed Forward Networks**

The second class of a Feed forward neural network distinguishes itself by the presence of one or more hidden layers, whose computational nodes are correspondingly called hidden neurons. The function of hidden neuron is to intervene between the external input and the network output in some useful manner. By adding more hidden layers, the network is enabled to extract higher order statistics. The input signal is applied to the neurons in the second layer. The output signal of second layer is used as inputs to the third layer, and so on for the rest of the network.

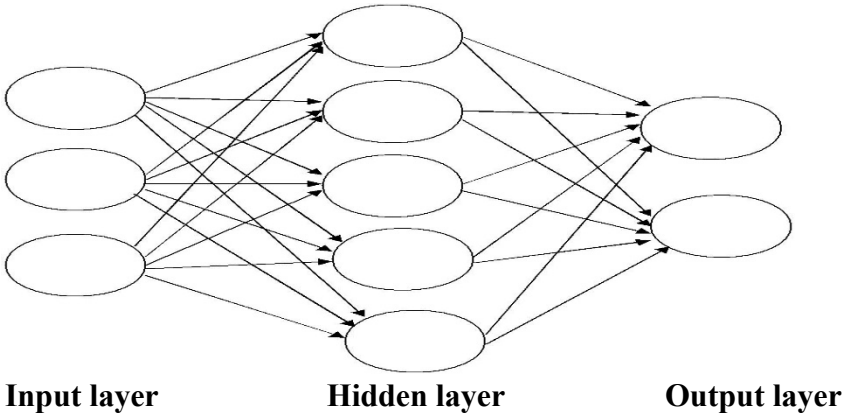

Fig. 5 Multilayer Feed Forward Network

3) **Recurrent networks**

A recurrent neural network has at least one feedback loop. A recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons. Self-feedback refers to a situation where the output of a neuron is fed back into its own input. The presence of feedback loops has a profound impact on the learning capability of the network and on its performance.

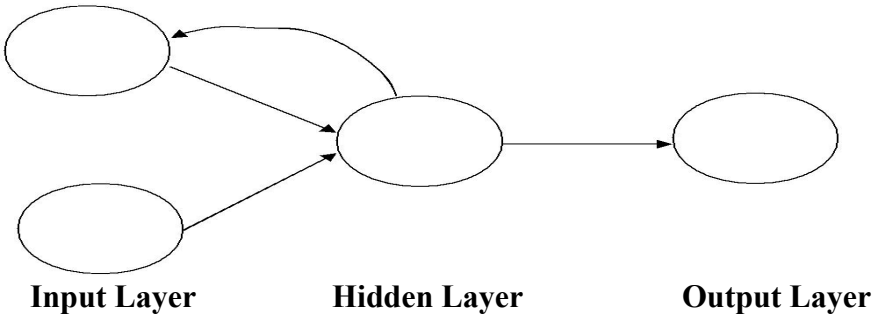

Fig. 6 Recurrent Networks

Network layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units.

- The activity of the input unit's represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single-layer and multi-layer architectures. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

A learning method used for adaptive neural networks can be classified into two major categories:

- Supervised learning which incorporates an external teacher, so that each output unit is told what its desired response to input signal sought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning. Important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. One well-known method, which is common to many learning paradigms, is the least mean square (LMS) convergence.
- Unsupervised learning uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian learning and competitive learning.

From Human Neurons to Artificial Neurons their aspect of learning concerns the distinction or not of a separate phase, during which the network is trained, and a subsequent operation phase. We say that a neural network learns of-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line; whereas unsupervised learning is performed on-line.

Activation Function:

The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (Activation function) that is specified for the units. This function typically falls into one of three categories:

- Linear (or ramp)
- Threshold
- Sigmoid

For linear units, the output activity is proportional to the total weighted output.

For threshold units, the output are set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

For sigmoid units, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

To make a neural network that performs some specific task, we must choose how the units are connected to one another, and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence.

We can teach a three-layer network to perform a particular task by using the following procedure:

1. We present the network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.
2. We determine how closely the actual output of the network matches the desired output.
3. We change the weight of each connections of the network produces a better approximation of the desired output.

II. Support Vector Machine:

In machine learning, **support vector machines** (**SVMs**, also **support vector networks**[1]) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p − 1)-dimensional hyper plane. This is called a linear classifier. There are many hyper planes that might classify the data. One reasonable choice as the best hyper plane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyper plane so that the distance from it to the nearest data point on each side is maximized. If such a hyper plane exists, it is known as the maximum-margin hyper plane and the linear classifier it defines is known as a maximum margin classifier; or equivalently, the perceptron of optimal stability

Linear SVM

Given some training data $\mathcal{D}$, a set of *n* points of the form

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}\}_{i=1}^{n}$$

Where the y_i is either 1 or −1, indicating the class to which the point $\mathbf{x}_i$ belongs. Each $\mathbf{x}_i$ is a *p*-dimensional real vector. We want to find the maximum-margin hyper plane that divides the points having $y_i = 1$ from those having $y_i = -1$. Any hyper plane can be written as the set of points $\mathbf{x}$ satisfying

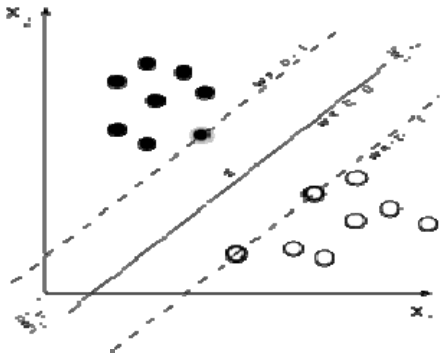

Fig. 7 Maximum-margin hyper plane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors

$$\mathbf{w} \cdot \mathbf{x} - b = 0,$$

Where $\cdot$ denotes the dot product and $\mathbf{w}$ the (not necessarily normalized) normal vector to the hyper plane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyper plane from the origin along the normal vector $\mathbf{w}$.

If the training data are linearly separable, we can select two hyper planes in a way that they separate the data and there are no points between them, and then try to maximize their distance. The region bounded by them is called "the margin". These hyper planes can be described by the equations

$$\mathbf{w} \cdot \mathbf{x} - b = 1$$

and

$$\mathbf{w} \cdot \mathbf{x} - b = -1.$$

By using geometry, we find the distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$, so we want to minimize $\|\mathbf{w}\|$. As we also have to prevent data points from falling into the margin, we add the following constraint: for each i either

$$\mathbf{w} \cdot \mathbf{x}_i - b \geq 1 \qquad \text{for } \mathbf{x}_i \text{ of the first class}$$

or

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1 \qquad \text{for } \mathbf{x}_i \text{ of the second.}$$

This can be rewritten as:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n. \qquad (1)$$

We can put this together to get the optimization problem:

Minimize (in $\mathbf{w}, b$)

$$\|\mathbf{w}\|$$

subject to (for any $i = 1, \ldots, n$)

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1.$$

Primal form

The optimization problem presented in the preceding section is difficult to solve because it depends on $\|\mathbf{w}\|$, the norm of $\mathbf{w}$, which involves a square root. Fortunately it is possible to alter the equation by substituting $\|\mathbf{w}\|$ with $\frac{1}{2}\|\mathbf{w}\|^2$ (the factor of 1/2 being used for mathematical convenience) without changing the solution (the minimum of the original and the modified equation have the same $\mathbf{w}$ and b). This is a quadratic programming optimization problem. More clearly:

$$\arg\min_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to (for any $i = 1, \ldots, n$)

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1.$$

By introducing Lagrange multipliers α, the previous constrained problem can be expressed as

$$\arg\min_{\mathbf{w},b} \max_{\alpha \geq 0} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\}$$

that is we look for a saddle point. In doing so all the points which can be separated as $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1 > 0$ do not matter since we must set the corresponding α_i to zero.

This problem can now be solved by standard quadratic programming techniques and programs. The "stationary" Karush–Kuhn–Tucker condition implies that the solution can be expressed as a linear combination of the training vectors

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i.$$

Only a few α_i will be greater than zero. The corresponding $\mathbf{x}_i$ are exactly the *support vectors*, which lie on the margin and satisfy $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) = 1$. From this one can derive that the support vectors also satisfy

$$\mathbf{w} \cdot \mathbf{x}_i - b = 1/y_i = y_i \iff b = \mathbf{w} \cdot \mathbf{x}_i - y_i$$

Which allows one to define the offset b. In practice, it is more robust to average over all N_{SV} support vectors:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (\mathbf{w} \cdot \mathbf{x}_i - y_i)$$

Multiclass SVM

Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements.

The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Common methods for such reduction include

- Building binary classifiers which distinguish between (i) one of the labels and the rest (*one-versus-all*) or (ii) between every pair of classes (*one-versus-one*). Classification of new instances for the one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class (it is important that the output functions be calibrated to produce comparable scores). For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with the most votes determines the instance classification.
- Directed acyclic graph SVM (DAGSVM)
- Error-correcting output codes

  Crammer and Singer proposed a multiclass SVM method which casts the multiclass classification problem into a single optimization problem, rather than decomposing it into multiple binary classification problems.

Transductive support vector machines

Transductive support vector machines extend SVMs in that they could also treat partially labeled data in semi-supervised learning by following the principles of transduction. Here, in addition to the training set $\mathcal{D}$, the learner is also given a set

$$\mathcal{D}^\star = \{\mathbf{x}_i^\star | \mathbf{x}_i^\star \in \mathbb{R}^p\}_{i=1}^k$$

of test examples to be classified. Formally, a transductive support vector machine is defined by the following primal optimization problem:

Minimize (in $\mathbf{w}, b, \mathbf{y}^\star$)

$$\frac{1}{2}\|\mathbf{w}\|^2$$

subject to (for any $i = 1, \ldots, n$ and any $j = 1, \ldots, k$)

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1,$$
$$y_j^\star(\mathbf{w} \cdot \mathbf{x}_j^\star - b) \geq 1,$$

and

$$y_j^\star \in \{-1, 1\}.$$

III. Decision Tree Learning:

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. More descriptive names for such tree models are **classification trees** or **regression trees**. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making. This page deals with decision trees in data mining.

Decision trees used in data mining are of two main types:

- **Classification tree** analysis is when the predicted outcome is the class to which the data belongs.
- **Regression tree** analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital).

  The term **Classification and Regression Tree (CART)** analysis is an umbrella term used to refer to both of the above procedures, first introduced by Breiman et al. Trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split.

  Some techniques, often called *ensemble* methods, construct more than one decision tree:

- **Bagging** decision trees, an early ensemble method, builds multiple decision trees by repeatedly resampling training data with replacement, and voting the trees for a consensus prediction.
- A **Random Forest** classifier uses a number of decision trees, in order to improve the classification rate.
- **Boosted Trees** can be used for regression-type and classification-type problems.
- **Rotation forest** - in which every decision tree is trained by first applying principal component analysis (PCA) on a random subset of the input features.

  Decision tree learning is the construction of a decision tree from class-labeled training tuples. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node.

Steps for Decision Tree Learning

Input: training set $\{(x_i, y_i)\}_{i=1}^n$; a differentiable loss function $L(y, F(x))$; number of iterations M.

Algorithm:

1) Initialize model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma).$$

2) For $m = 1$ to M:
   a) Compute so-called *pseudo-residuals*:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

   b) Fit a base learner $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^n$.
   c) Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right).$$

   d) Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3) Output $F_M(x)$.

TIME LINE

Sr. No.	Topics	Time Taken
1.	Course work and Result	Six to Eight Months
2.	Literature Survey and Synopsis Presentation	Four to Six Months
3.	Learning Machine Learning Tools	Five to Eight Months
4.	Implementation	Three to Eight Months
5.	Comparison of Results	Three to Four Months
6.	Preparation of Research Papers in referred journals	Eight to Twelve Months
7.	Report Preparation	Four to Six months

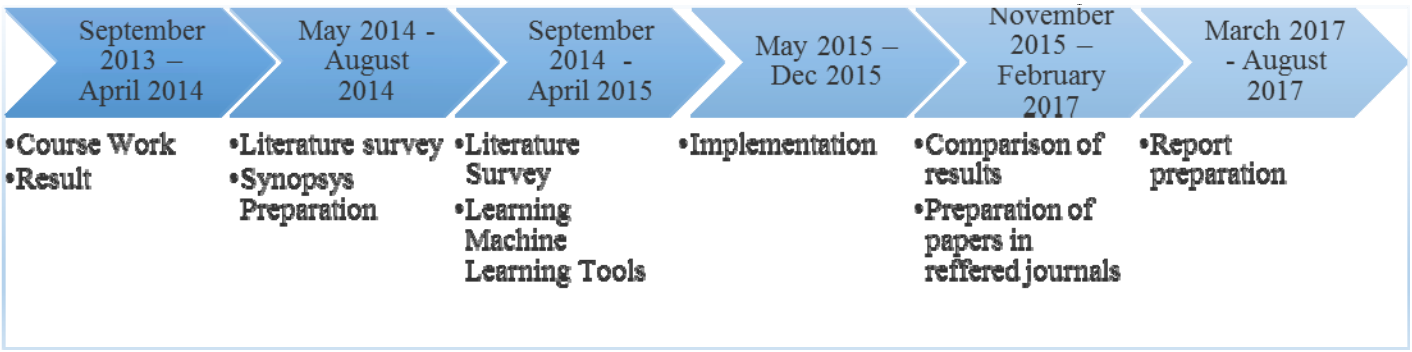

Fig. 8 Time Line for Research Work

References

[1]. www.wikipedia.org

[2]. http://www.kaggle.com/

[3]. Berges, M., Goldman, E., Matthews, H.S., and Soibelman, L. Training Load Monitoring Algorithms on Highly Sub-Metered Home Electricity Consumption Data. *Tsinghua Science & Technology 13, Supple*, 0 (2008), 406–411.

[4]. Carrie Armel, K., Gupta, A., Shrimali, G., and Albert, A. Is disaggregation the holy grail of energy efficiency? *The case of electricity. Energy Policy* 52, (2012), 213–234.

[5]. Froehlich, J., Larson, E., Gupta, S., Cohn, G., Reynolds, M., and Patel, S. Disaggregated End-Use Energy Sensing for the Smart Grid. *IEEE Pervasive Computing 10*, 1 (2011), 28–39.

[6]. Gupta, S., Reynolds, M., and Patel, S. ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the home. *Ubicomp 2010*, (2010).

[7]. Laughman, C., Lee, K., and Cox, R. Power signature analysis. *IEEE Power and Energy*, april 2003 (2003).

[8]. Zeifman, M., Ph, D., and Roth, K. Non-Intrusive Appliance Load Monitoring (NIALM): Review and Outlook * Fraunhofer : A Leading Force in Applied R & D. Consumer Electronics, January (2011).

[9]. Rudin, C.; Waltz, D.; Anderson, R.N.; Boulanger, A; Salleb-Aouissi, A; Chow, M.; Dutta, H.; Gross, P.N.; Huang, B.; Ierome, S.; Isaac, D.F.; Kressner, A; Passonneau, R.J.; Radeva, A; Wu, L., "Machine Learning for the New York City Power Grid," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.34, no.2, pp.328-345, Feb. 2012.

[10]. Hassan, T.; Javed, F.; Arshad, N., "An Empirical Investigation of V-I Trajectory Based Load Signatures for Non-Intrusive Load Monitoring," Smart Grid, IEEE Transactions on , vol.5, no.2, pp.870 - 878, March 2014.

[11]. Figueiredo, M.; Ribeiro, B.; de Almeida, A, "Electrical Signal Source Separation Via Nonnegative Tensor Factorization Using On Site Measurements in a Smart Home," Instrumentation and Measurement, IEEE Transactions on , vol.63, no.2, pp.364- 373, Feb. 2014.

[12]. Qinran Hu; Fangxing Li, "Hardware Design of Smart Home Energy Management System With Dynamic Price Response," Smart Grid, IEEE Transactions on , vol.4, no.4, pp.1878 - 1887, Dec. 2013.

[13]. Srinivasarengan, K.; Goutam, Y.G.; Chandra, M.G.; Kadhe, S., "A Framework for Non-Intrusive Load Monitoring Using Bayesian Inference," Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on , vol., no., pp.427 - 432, 3-5 July 2013.

[14]. Chaouachi, A; Kamel, R.M.; Andoulsi, R.; Nagasaka, K., "Multiobjective Intelligent Energy Management for a Microgrid," Industrial Electronics, IEEE Transactions on, vol.60, no.4, pp.1688 - 1699, April 2013.

[15]. Hongming Yang; Pandharipande, A, "Disaggregated Daylight–Artificial Light Sensing," Sensors Journal, IEEE, vol.12, no.12, pp.3438 - 3445, Dec. 2012.

[16]. Mishra, A; Irwin, D.; Shenoy, P.; Kurose, J.; Ting Zhu, "SmartCharge: Cutting the electricity bill in smart homes with energy storage," Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on , vol., no., pp.1 - 10, 9-11 May 2012.

[17]. Niyato, D.; Lu Xiao; Ping Wang, "Machine-to-machine communications for home energy management system in smart grid," Communications Magazine, IEEE, vol.49, no.4, pp.53 - 59, April 2011.

[18]. Filippi, A; Pandharipande, A; Lelkens, A; Rietman, R.; Schenk, T.; Ying Wang; Shrubsole, P., "Multi-appliance power disaggregation: An approach to energy monitoring," Energy Conference and Exhibition (EnergyCon), 2010 IEEE International, vol., no., pp.91 - 95, 18-22 Dec. 2010.

[19]. Dhiman, G.; Rosing, T.S., "Dynamic Power Management Using Machine Learning," Computer-Aided Design, 2006. ICCAD '06. IEEE/ACM International Conference on, vol., no., pp.747 - 754, 5-9 Nov. 2006.

[20]. Xia Chen; Zhao Yang Dong; Ke Meng; Yan Xu; Kit Po Wong; Ngan, H. W., "Electricity Price Forecasting With Extreme Learning Machine and Bootstrapping," Power Systems, IEEE Transactions on , vol.27, no.4, pp.2055 - 2062, Nov. 2012.

[21]. Bo-Juen Chen; Ming-Wei Chang; Chih-Jen Lin, "Load forecasting using support vector Machines: a study on EUNITE competition 2001," Power Systems, IEEE Transactions on, vol.19, no.4, pp.1821-1830, Nov. 2004.

[22]. Can Wan; Zhao Xu; Pinson, P.; Zhao Yang Dong; Kit Po Wong, "Probabilistic Forecasting of Wind Power Generation Using Extreme Learning Machine," Power Systems, IEEE Transactions on , vol.29, no.3, pp.1033 -1044, May 2014.

[23]. Sharma, N.; Sharma, P.; Irwin, D.; Shenoy, P., "Predicting solar generation from weather forecasts using machine learning," Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on , vol., no., pp.528 - 533, 17-20 Oct. 2011.

[24]. Jinyu Tian; Yan Lin, "Short-Term Electricity Price Forecasting Based on Rough Sets and Improved SVM," Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on, vol., no., pp.68 - 71, 23-25 Jan. 2009.

[25]. Fan, S.; Mao, C.; Chen, L., "Next-day electricity-price forecasting using a hybrid network," Generation, Transmission & Distribution, IET, vol.1, no.1, pp.176 - 182, January 2007.

[26]. Yaochu Jin; Sendhoff, B., "Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol.38, no.3, pp.397 - 415, May 2008.

[27]. Sapankevych, N.I; Sankar, Ravi, "Time Series Prediction Using Support Vector Machines: A Survey," Computational Intelligence Magazine, IEEE, vol.4, no.2, pp.24 - 38, May 2009.

[28]. Sarpeshkar, R.; Wattanapanitch, W.; Rapoport, B.I; Arfin, S.K.; Baker, M.W.; Mandal, S.; Fee, M.S.; Musallam, S.; Andersen, R.A, "Low-Power Circuits for Brain-Machine Interfaces," Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on, vol., no., pp.2068, 2071, 27-30 May 2007.

[29]. Kalyani, S.; Swarup, K.S., "Classification and Assessment of Power System Security Using Multiclass SVM," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol.41, no.5, pp.753 - 758, Sept. 2011.

[30]. Haoyang Shen; Hino, H.; Murata, N.; Wakao, S., "A Measure of Credibility of Solar Power Prediction," Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, vol.2, no., pp.286 - 291, 18-21 Dec. 2011.

[31]. Jiang Chang; Shu-Yun Jia, "Windy-solar Power Generation system based on Multi-agent System," Machine Learning and Cybernetics, 2008 International Conference on, vol.4, no., pp.2446 - 2449, 12-15 July 2008.

[32]. Can Wan; Zhao Xu; Pinson, P.; Zhao Yang Dong; Kit Po Wong, "Optimal Prediction Intervals of Wind Power Generation," Power Systems, IEEE Transactions on , vol.29, no.3, pp.1166 - 1174, May 2014.

[33]. Yan Xu; Zhao-Yang Dong; Zhao Xu; Ke Meng; Kit Po Wong, "An Intelligent Dynamic Security Assessment Framework for Power Systems With Wind Power," Industrial Informatics, IEEE Transactions on , vol.8, no.4, pp.995 - 1003, Nov. 2012.

[34]. Yan Xu; Zhao Yang Dong; Jun Hua Zhao; Pei Zhang; Kit Po Wong, "A Reliable Intelligent System for Real-Time Dynamic Security Assessment of Power Systems," Power Systems, IEEE Transactions on , vol.27, no.3, pp.1253 - 1263, Aug. 2012.